Approximate Maximum Likelihood Estimation for Linear Regression with Operator Uncertainty

Richard Clancy
(joint with Stephen Becker)

University of Colorado at Boulder
Department of Applied Mathematics

March 13th
FRAMSC 2021

Outline

Table of Contents

Problem Setup

We consider the generative model $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x} + \boldsymbol{\eta}$

- $\boldsymbol{G} \in \mathbb{R}^{m \times n}$ is a random matrix
- $\boldsymbol{\eta} \in \mathbb{R}^{m}$ is a random vector
- $\boldsymbol{x} \in \mathbb{R}^{n}$ is vector of model parameters
- $\boldsymbol{y} \in \mathbb{R}^{m}$ is a vector of measurements

Problem Setup

We consider the generative model $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x} + \boldsymbol{\eta}$

- $\boldsymbol{G} \in \mathbb{R}^{m \times n}$ is a random matrix
- $\boldsymbol{\eta} \in \mathbb{R}^{m}$ is a random vector
- $\boldsymbol{x} \in \mathbb{R}^{n}$ is vector of model parameters
- $\boldsymbol{y} \in \mathbb{R}^{m}$ is a vector of measurements

Given measurement vector $\boldsymbol{y}$ and distributional knowledge of $\boldsymbol{G}$ and $\boldsymbol{\eta}$, estimate $\boldsymbol{x}$.

Why should we care?

In practice, it is uncommon to know $\boldsymbol{G}$ precisely. Some causes are

- precision limits in measurement
- truncation error for memory savings
- sampling error
- human error
- modeling error

Estimating home prices

Generative model $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x} + \boldsymbol{\eta}$

Estimating home prices

Generative model $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x} + \boldsymbol{\eta}$

Example

- $\boldsymbol{G} \in \mathbb{R}^{m \times 2}$, first column is square footage of home, second is square footage of lot. We observe $\boldsymbol{H} = \text{round}(\boldsymbol{G})$ to nearest hundred foot (e.g. 1366 $\rightarrow$ 1400 sqft house)

Estimating home prices

Generative model $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x} + \boldsymbol{\eta}$

Example

- $\boldsymbol{G} \in \mathbb{R}^{m \times 2}$, first column is square footage of home, second is square footage of lot. We observe $\boldsymbol{H} = \text{round}(\boldsymbol{G})$ to nearest hundred foot (e.g. 1366 $\rightarrow$ 1400 sqft house)
- G_{ij} can be modeled as a Uniform$(H_{ij} - \delta, H_{ij} + \delta)$ for $\delta = 50$ and $H_{ij} = 1400$

Estimating home prices

Generative model $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x} + \boldsymbol{\eta}$

Example

- $\boldsymbol{G} \in \mathbb{R}^{m \times 2}$, first column is square footage of home, second is square footage of lot. We observe $\boldsymbol{H} = \text{round}(\boldsymbol{G})$ to nearest hundred foot (e.g. 1366 $\rightarrow$ 1400 sqft house)
- G_{ij} can be modeled as a Uniform$(H_{ij} - \delta, H_{ij} + \delta)$ for $\delta = 50$ and $H_{ij} = 1400$
- $\boldsymbol{y} \in \mathbb{R}^m$ is selling price for corresponding home (e.g. \$207k)

Estimating home prices

Generative model $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x} + \boldsymbol{\eta}$

Example

- $\boldsymbol{G} \in \mathbb{R}^{m \times 2}$, first column is square footage of home, second is square footage of lot. We observe $\boldsymbol{H} = \text{round}(\boldsymbol{G})$ to nearest hundred foot (e.g. 1366 $\to$ 1400 sqft house)
- G_{ij} can be modeled as a $\text{Uniform}(H_{ij} - \delta, H_{ij} + \delta)$ for $\delta = 50$ and $H_{ij} = 1400$
- $\boldsymbol{y} \in \mathbb{R}^{m}$ is selling price for corresponding home (e.g. \$207k)
- GOAL: Estimate parameters $\boldsymbol{x}$ so we can model price based on home and lot size accounting for uncertainty in $\boldsymbol{G}$ and $\boldsymbol{\eta}$

Table of Contents

Maximum likelihood estimation

- Given observed data $\boldsymbol{y}$ and a likelihood function $L(\boldsymbol{x}) = f_{\boldsymbol{Y}}(\boldsymbol{y}; \boldsymbol{x})$, where $f_{\boldsymbol{Y}}(\boldsymbol{y}; \boldsymbol{x})$ is the PDF of $\boldsymbol{y}$, find parameters $\boldsymbol{x}$ that maximize the likelihood function,

$$\operatorname{argmax}_{\boldsymbol{x}} L(\boldsymbol{x})$$

Maximum likelihood estimation

- Given observed data $\boldsymbol{y}$ and a likelihood function $L(\boldsymbol{x}) = f_{\boldsymbol{Y}}(\boldsymbol{y}; \boldsymbol{x})$, where $f_{\boldsymbol{Y}}(\boldsymbol{y}; \boldsymbol{x})$ is the PDF of $\boldsymbol{y}$, find parameters $\boldsymbol{x}$ that maximize the likelihood function,

$$\mathrm{argmax}_{\boldsymbol{x}} L(\boldsymbol{x})$$

- When components of $\boldsymbol{y}$ are independent, we can split PDF such that

$$f_{\boldsymbol{Y}}(\boldsymbol{y}; \boldsymbol{x}) = \prod_{i=1}^{m} f_{Y_i}(y_i; \boldsymbol{x})$$

Maximum likelihood estimation

- Given observed data $\boldsymbol{y}$ and a likelihood function $L(\boldsymbol{x}) = f_{\boldsymbol{Y}}(\boldsymbol{y}; \boldsymbol{x})$, where $f_{\boldsymbol{Y}}(\boldsymbol{y}; \boldsymbol{x})$ is the PDF of $\boldsymbol{y}$, find parameters $\boldsymbol{x}$ that maximize the likelihood function,

$$\mathrm{argmax}_{\boldsymbol{x}} L(\boldsymbol{x})$$

- When components of $\boldsymbol{y}$ are independent, we can split PDF such that

$$f_{\boldsymbol{Y}}(\boldsymbol{y}; \boldsymbol{x}) = \prod_{i=1}^{m} f_{Y_i}(y_i; \boldsymbol{x})$$

- We focus on maximizing the log-likelihood function

$$\hat{\boldsymbol{x}}_{MLE} = \mathrm{argmax}_{\boldsymbol{x}} \left\{ \sum_{i=1}^{m} \ln \left[f_{Y_i}(y_i; \boldsymbol{x}) \right] \right\}$$

Justification of MLE for regression problem

For additive noise models $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\eta}$ where $\boldsymbol{H}$ is known precisely (all uncertainty is in $\boldsymbol{\eta}$), we have

Justification of MLE for regression problem

For additive noise models $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\eta}$ where $\boldsymbol{H}$ is known precisely (all uncertainty is in $\boldsymbol{\eta}$), we have

- $\text{argmin}_{\boldsymbol{x}} \|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\|_2^2$ is the MLE for Gaussian noise (ordinary least squares)

Justification of MLE for regression problem

For additive noise models $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\eta}$ where $\boldsymbol{H}$ is known precisely (all uncertainty is in $\boldsymbol{\eta}$), we have

- $\operatorname{argmin}_{\boldsymbol{x}} \|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\|_2^2$ is the MLE for Gaussian noise (ordinary least squares)
- $\operatorname{argmin}_{\boldsymbol{x}} \|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\|_1$ is the MLE for double exponential noise

Justification of MLE for regression problem

For additive noise models $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\eta}$ where $\boldsymbol{H}$ is known precisely (all uncertainty is in $\boldsymbol{\eta}$), we have

- $\operatorname{argmin}_{\boldsymbol{x}} \|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\|_2^2$ is the MLE for Gaussian noise (ordinary least squares)
- $\operatorname{argmin}_{\boldsymbol{x}} \|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\|_1$ is the MLE for double exponential noise
- $\operatorname{argmin}_{\boldsymbol{x}} \|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\|_\infty$ is the MLE for uniform noise.

Justification of MLE for regression problem

For additive noise models $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\eta}$ where $\boldsymbol{H}$ is known precisely (all uncertainty is in $\boldsymbol{\eta}$), we have

- $\text{argmin}_{\boldsymbol{x}} \|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\|_2^2$ is the MLE for Gaussian noise (ordinary least squares)
- $\text{argmin}_{\boldsymbol{x}} \|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\|_1$ is the MLE for double exponential noise
- $\text{argmin}_{\boldsymbol{x}} \|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\|_\infty$ is the MLE for uniform noise.
- For uncertainty in operator, total least squares, i.e.,

$$\begin{aligned} \min_{\boldsymbol{x}, \boldsymbol{U}, \boldsymbol{\eta}} \quad & \|[\boldsymbol{U}, \boldsymbol{\eta}]\|_F \\ \text{Subject to} \quad & (\boldsymbol{H} + \boldsymbol{U})\boldsymbol{x} = \boldsymbol{y} + \boldsymbol{\eta} \end{aligned}$$

  is the MLE for i.i.d. Gaussian in $\boldsymbol{H}$ and $\boldsymbol{\eta}$

Justification of MLE for regression problem

For additive noise models $\boldsymbol{y} = \boldsymbol{Hx} + \boldsymbol{\eta}$ where $\boldsymbol{H}$ is known precisely (all uncertainty is in $\boldsymbol{\eta}$), we have

- $\text{argmin}_{\boldsymbol{x}} \|\boldsymbol{Hx} - \boldsymbol{y}\|_2^2$ is the MLE for Gaussian noise (ordinary least squares)
- $\text{argmin}_{\boldsymbol{x}} \|\boldsymbol{Hx} - \boldsymbol{y}\|_1$ is the MLE for double exponential noise
- $\text{argmin}_{\boldsymbol{x}} \|\boldsymbol{Hx} - \boldsymbol{y}\|_\infty$ is the MLE for uniform noise.
- For uncertainty in operator, total least squares, i.e.,

$$\begin{array}{cc} \min\limits_{\boldsymbol{x}, \boldsymbol{U}, \boldsymbol{\eta}} & \|[\boldsymbol{U}, \eta]\|_F \\ \text{Subject to} & (\boldsymbol{H} + \boldsymbol{U})\boldsymbol{x} = \boldsymbol{y} + \boldsymbol{\eta} \end{array}$$

is the MLE for i.i.d. Gaussian in $\boldsymbol{H}$ and $\boldsymbol{\eta}$

MLE for uncertainty in design matrix

- We seek a PDF for $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x} + \boldsymbol{\eta}$ to form a likelihood function

MLE for uncertainty in design matrix

- We seek a PDF for $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x} + \boldsymbol{\eta}$ to form a likelihood function
- Each component can be rewritten as a sum of RVs, i.e., $y_i = \sum_{j=1}^{n} G_{ij} x_j + \eta_i$

MLE for uncertainty in design matrix

- We seek a PDF for $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x} + \boldsymbol{\eta}$ to form a likelihood function
- Each component can be rewritten as a sum of RVs, i.e., $y_i = \sum_{j=1}^{n} G_{ij}x_j + \eta_i$
- PDFs for sums of RVs are challenging to derive since they generally require convolutions

MLE for uncertainty in design matrix

- We seek a PDF for $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x} + \boldsymbol{\eta}$ to form a likelihood function
- Each component can be rewritten as a sum of RVs, i.e., $y_i = \sum_{j=1}^{n} G_{ij}x_j + \eta_i$
- PDFs for sums of RVs are challenging to derive since they generally require convolutions

Example

Let $Z \sim \mathcal{N}(0,1)$ and $U \sim \text{Uniform}(0,1)$. The PDF for $U + Z$ is

$$f_{U+Z}(t) = \frac{1}{\sqrt{2\pi}} \int_0^1 e^{-(t-s)^2/2}\, ds.$$

No analytic form!

Table of Contents

Moment generating functions to the rescue

- The moment generating function (MGF) is a bilateral Laplace transform of PDF given by

$$M_Y(t) = \mathbb{E}(e^{tY})$$

Moment generating functions to the rescue

- The moment generating function (MGF) is a bilateral Laplace transform of PDF given by

$$M_Y(t) = \mathbb{E}(e^{tY})$$

- Useful properties, i.e., for indep. RVs $U, \ Z$ and $a \in \mathbb{R}$

$$M_{aU+Z}(t) = M_U(at)M_Z(t)$$

Moment generating functions to the rescue

- The moment generating function (MGF) is a bilateral Laplace transform of PDF given by

$$M_Y(t) = \mathbb{E}(e^{tY})$$

- Useful properties, i.e., for indep. RVs U, Z and $a \in \mathbb{R}$

$$M_{aU+Z}(t) = M_U(at) M_Z(t)$$

- When MGF exists, it uniquely characterizes the distribution

Moment generating functions to the rescue

- The moment generating function (MGF) is a bilateral Laplace transform of PDF given by

$$M_Y(t) = \mathbb{E}(e^{tY})$$

- Useful properties, i.e., for indep. RVs $U, \ Z$ and $a \in \mathbb{R}$

$$M_{aU+Z}(t) = M_U(at)M_Z(t)$$

- When MGF exists, it uniquely characterizes the distribution
- With MGFs, convolution $\rightarrow$ multiplication

Moment generating functions to the rescue

- The moment generating function (MGF) is a bilateral Laplace transform of PDF given by

$$M_Y(t) = \mathbb{E}(e^{tY})$$

- Useful properties, i.e., for indep. RVs $U,\ Z$ and $a \in \mathbb{R}$

$$M_{aU+Z}(t) = M_U(at)M_Z(t)$$

- When MGF exists, it uniquely characterizes the distribution
- With MGFs, convolution $\rightarrow$ multiplication

Example

Let $Z \sim \mathcal{N}(0,1)$ and $U \sim \text{Uniform}(0,1)$. The MGF for $U + Z$ is

$$M_{U+Z}(t) = \frac{(e^t - 1)e^{-t^2/2}}{t}.$$

Analytic form, no need for quadrature

Moment and generating functions

- Using properties of MGFs, we have

$$M_{Y_i}(t) = M_{\boldsymbol{g}_i^T \boldsymbol{x} + \eta_i}(t) = M_{\eta_i}(t) \prod_{j=1}^{n} M_{G_{ij}}(t x_j)$$

Moment and generating functions

- Using properties of MGFs, we have

$$M_{Y_i}(t) = M_{\boldsymbol{g}_i^T \boldsymbol{x} + \eta_i}(t) = M_{\eta_i}(t) \prod_{j=1}^{n} M_{G_{ij}}(t x_j)$$

- Importantly, we have expressed a complicated MGF for Y_i as the product of simple univariate MGFs for G_{ij} and η_i.

Moment and generating functions

- Using properties of MGFs, we have

$$M_{Y_i}(t) = M_{\boldsymbol{g}_i^T \boldsymbol{x} + \eta_i}(t) = M_{\eta_i}(t) \prod_{j=1}^{n} M_{G_{ij}}(t x_j)$$

- Importantly, we have expressed a complicated MGF for Y_i as the product of simple univariate MGFs for G_{ij} and η_i.
- By inverting transform for M_{Y_i}, we can recover density, but difficult in practice

Moment and generating functions

- Using properties of MGFs, we have

$$M_{Y_i}(t) = M_{\boldsymbol{g}_i^T \boldsymbol{x} + \eta_i}(t) = M_{\eta_i}(t) \prod_{j=1}^{n} M_{G_{ij}}(t x_j)$$

- Importantly, we have expressed a complicated MGF for Y_i as the product of simple univariate MGFs for G_{ij} and η_i.
- By inverting transform for M_{Y_i}, we can recover density, but difficult in practice
- Use approximation method instead!

Density approximation

Using MGFs to approximate PDFs allows for construction of a likelihood. Some options are

- Edgeworth series [1]: poor tail behavior (polynomial series)
- Kernel density estimation [2, 3]: data intensive
- Saddle point approximation [4, 5, 6]: uses exponential tilting and works well in practice

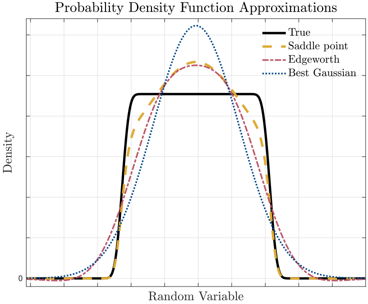

Figure: True density and several approximations when $y = \boldsymbol{g}^T \boldsymbol{x} + \eta$

Saddle point approximation

Saddle point approximation for PDF of RV Y is

$$f_Y(y) \approx \sqrt{\frac{1}{2\pi K_Y''(t_0)}} e^{K_Y(t_0) - y t_0}$$

- $K_Y(t) = \ln M_Y(t)$ is *Cumulant Generating Function* (CGF)
- t_0 is the solution to $K_Y'(t) - y = 0$ (use Newton's method)

Table of Contents

Approximate MLE

Using the saddle point approximation and eliminating constants allows us to write the approximate log-likelihood function as

$$\begin{aligned}\ell(\boldsymbol{x}) &= \sum_{i=1}^{m} \ln\left\{\sqrt{\frac{1}{K''_{Y_i}(t_i)}} e^{K_{Y_i}(t_i) - y_i t_i}\right\} \\ &= \sum_{i=1}^{m} \left\{K_{Y_i}(t_i) - t_i y_i - \frac{1}{2}\ln\left(K''_{Y_i}(t_i)\right)\right\} \\ &= \sum_{i=1}^{m} \left[K_{\boldsymbol{g}_i^T \boldsymbol{x} + \eta_i}(t_i(\boldsymbol{x})) - \frac{1}{2}\ln\left(K''_{\boldsymbol{g}_i^T \boldsymbol{x} + \eta_i}(t_i(\boldsymbol{x}))\right) - t_i(\boldsymbol{x}) y_i\right].\end{aligned}$$

where $\boldsymbol{g}_i^T$ is i^{th} row of $\boldsymbol{G}$ and $t_i(\boldsymbol{x})$ is solution to $K'_{\boldsymbol{g}_i^T \boldsymbol{x} + \eta_i}(t) = y_i$.

Optimization problem

The approximate MLE can be cast generically in vector form

$$\begin{aligned} \text{argmax}_{\boldsymbol{x},\boldsymbol{t}} \quad & \mathbb{1}^T \left(K_{\boldsymbol{Gx}+\boldsymbol{\eta}}(\boldsymbol{t}) - \frac{1}{2} \ln \left(K''_{\boldsymbol{Gx}+\boldsymbol{\eta}}(\boldsymbol{t}) \right) \right) - \boldsymbol{t}^T \boldsymbol{y} \\ \text{Subject to} \quad & K'_{\boldsymbol{Gx}+\boldsymbol{\eta}}(\boldsymbol{t}) = \boldsymbol{y} \end{aligned}$$

Optimization problem

The approximate MLE can be cast generically in vector form

$$\begin{aligned} \operatorname{argmax}_{\boldsymbol{x},\boldsymbol{t}} \quad & \mathbb{1}^T \left(K_{\boldsymbol{Gx}+\boldsymbol{\eta}}(\boldsymbol{t}) - \frac{1}{2} \ln \left(K''_{\boldsymbol{Gx}+\boldsymbol{\eta}}(\boldsymbol{t}) \right) \right) - \boldsymbol{t}^T \boldsymbol{y} \\ \text{Subject to} \quad & K'_{\boldsymbol{Gx}+\boldsymbol{\eta}}(\boldsymbol{t}) = \boldsymbol{y} \end{aligned}$$

Example

When $\boldsymbol{G} \sim \text{Uniform}(\boldsymbol{H} - \delta \mathbb{1}\mathbb{1}^T, \boldsymbol{H} + \delta \mathbb{1}\mathbb{1}^T)$ and $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$,

$$\begin{aligned} \operatorname{argmax}_{\boldsymbol{x},\boldsymbol{t}} \quad & \boldsymbol{t}^T \left(\frac{\sigma^2}{2} \boldsymbol{t} + \boldsymbol{H} \boldsymbol{x} - \boldsymbol{y} \right) + \mathbb{1}^T \ln \left[\sinh \left(\delta \boldsymbol{t} \boldsymbol{x}^T \right) \oslash \left(\delta \boldsymbol{t} \boldsymbol{x}^T \right) \right] \mathbb{1} \\ & - \frac{1}{2} \mathbb{1}^T \ln \left[\sigma^2 \mathbb{1} - \delta^2 \operatorname{csch}^2 \left(\delta \boldsymbol{t} \boldsymbol{x}^T \right) \boldsymbol{x}^2 \right] \end{aligned}$$

$$\text{Subject to} \qquad \sigma^2 \boldsymbol{t} + \boldsymbol{H}\boldsymbol{x} + \delta \coth \left(\delta \boldsymbol{t} \boldsymbol{x}^T \right) \boldsymbol{x} - n(\mathbb{1} \oslash \boldsymbol{t}) = \boldsymbol{y}.$$

Gradients

- Despite constraint requiring a numeric solve, have gradients

Gradients

- Despite constraint requiring a numeric solve, have gradients
- Letting $\boldsymbol{q}(\boldsymbol{x}, \boldsymbol{t}) = K'_{\boldsymbol{G}\boldsymbol{x}+\boldsymbol{\eta}}(\boldsymbol{t}) - \boldsymbol{y}$ be our constraint, then using adjoint state method [7], we have

$$\nabla_{\boldsymbol{x}} \ell = \frac{\partial \ell}{\partial \boldsymbol{x}} - \left(\frac{\partial \ell}{\partial \boldsymbol{t}}\right) \left(\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{t}}\right)^{-1} \left(\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}}\right). \tag{1}$$

Gradients

- Despite constraint requiring a numeric solve, have gradients
- Letting $\boldsymbol{q}(\boldsymbol{x}, \boldsymbol{t}) = K'_{\boldsymbol{Gx}+\boldsymbol{\eta}}(\boldsymbol{t}) - \boldsymbol{y}$ be our constraint, then using adjoint state method [7], we have

$$\nabla_{\boldsymbol{x}} \ell = \frac{\partial \ell}{\partial \boldsymbol{x}} - \left(\frac{\partial \ell}{\partial \boldsymbol{t}} \right) \left(\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{t}} \right)^{-1} \left(\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}} \right) . \tag{1}$$

Unimportant, but for completeness, each factor is given by:

$$\begin{aligned}
\frac{\partial \ell}{\partial \boldsymbol{x}} &= \mathbb{1}^T \left(\frac{\partial}{\partial \boldsymbol{x}} K_{\boldsymbol{Gx}+\boldsymbol{\eta}}(\boldsymbol{t}) - \frac{1}{2} \left\{ \operatorname{diag}\left(K''_{\boldsymbol{Gx}+\boldsymbol{\eta}}(\boldsymbol{t}) \right) \right\}^{-1} \frac{\partial}{\partial \boldsymbol{x}} K''_{\boldsymbol{Gx}+\boldsymbol{\eta}}(\boldsymbol{t}) \right) \\
\frac{\partial \ell}{\partial \boldsymbol{t}} &= \left(K'_{\boldsymbol{Gx}+\boldsymbol{\eta}}(\boldsymbol{t}) - \frac{1}{2} \left(K'''_{\boldsymbol{Gx}+\boldsymbol{\eta}}(\boldsymbol{t}) \oslash K''_{\boldsymbol{Gx}+\boldsymbol{\eta}}(\boldsymbol{t}) \right) - \boldsymbol{y} \right)^T \\
\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{t}} &= \operatorname{diag}\left(K''_{\boldsymbol{Gx}+\boldsymbol{\eta}}(\boldsymbol{t}) \right) , \\
\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{x}} &= \frac{\partial}{\partial \boldsymbol{x}} K'_{\boldsymbol{Gx}+\boldsymbol{\eta}}(\boldsymbol{t}).
\end{aligned}$$

Table of Contents

Algorithms and Numerical Experiments

- With gradient information, we can employ first order methods

Algorithms and Numerical Experiments

- With gradient information, we can employ first order methods
- For experiments, we opted for L-BFGS [8], a quasi-Newton method, which solved problem rapidly

Algorithms and Numerical Experiments

- With gradient information, we can employ first order methods
- For experiments, we opted for L-BFGS [8], a quasi-Newton method, which solved problem rapidly
- Although possible to calculate derivatives analytically in many cases, automatic differentiation can save time and trouble [9]

- To validate our method, we solved 10,000 problems. Using the generative model $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x}_{\mathsf{TRU}} + \boldsymbol{\eta}$ with components drawn as follows:

Numerical Experiments

- To validate our method, we solved 10,000 problems. Using the generative model $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x}_{\mathrm{TRU}} + \boldsymbol{\eta}$ with components drawn as follows:
  - $\boldsymbol{G} \in \mathbb{R}^{m \times n}$, from continuous uniform matrices on $[0, 10]$

Numerical Experiments

- To validate our method, we solved 10,000 problems. Using the generative model $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x}_{\text{TRU}} + \boldsymbol{\eta}$ with components drawn as follows:
  - $\boldsymbol{G} \in \mathbb{R}^{m \times n}$, from continuous uniform matrices on $[0, 10]$
  - $\boldsymbol{x}_{\text{TRU}} \in \mathbb{R}^{n}$, taken from the heavy-tailed Cauchy distribution to make the use of prior information on the solution difficult

Numerical Experiments

- To validate our method, we solved 10,000 problems. Using the generative model $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x}_{\text{TRU}} + \boldsymbol{\eta}$ with components drawn as follows:
  - $\boldsymbol{G} \in \mathbb{R}^{m \times n}$, from continuous uniform matrices on $[0, 10]$
  - $\boldsymbol{x}_{\text{TRU}} \in \mathbb{R}^n$, taken from the heavy-tailed Cauchy distribution to make the use of prior information on the solution difficult
  - $\boldsymbol{\eta} \in \mathbb{R}^m$ has i.i.d. components drawn from a normal distribution, i.e., $\eta_i \sim \mathcal{N}(0, \sigma^2)$

Numerical Experiments

- To validate our method, we solved 10,000 problems. Using the generative model $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x}_{\text{TRU}} + \boldsymbol{\eta}$ with components drawn as follows:
  - $\boldsymbol{G} \in \mathbb{R}^{m \times n}$, from continuous uniform matrices on $[0, 10]$
  - $\boldsymbol{x}_{\text{TRU}} \in \mathbb{R}^{n}$, taken from the heavy-tailed Cauchy distribution to make the use of prior information on the solution difficult
  - $\boldsymbol{\eta} \in \mathbb{R}^{m}$ has i.i.d. components drawn from a normal distribution, i.e., $\eta_i \sim \mathcal{N}(0, \sigma^2)$
- Assuming knowledge of σ^2, and inferred value of δ by observing $\boldsymbol{H} = \text{round}(\boldsymbol{G})$ and $\boldsymbol{y}$

Numerical Experiments

- To validate our method, we solved 10,000 problems. Using the generative model $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x}_{\text{TRU}} + \boldsymbol{\eta}$ with components drawn as follows:
  - $\boldsymbol{G} \in \mathbb{R}^{m \times n}$, from continuous uniform matrices on $[0, 10]$
  - $\boldsymbol{x}_{\text{TRU}} \in \mathbb{R}^{n}$, taken from the heavy-tailed Cauchy distribution to make the use of prior information on the solution difficult
  - $\boldsymbol{\eta} \in \mathbb{R}^{m}$ has i.i.d. components drawn from a normal distribution, i.e., $\eta_i \sim \mathcal{N}(0, \sigma^2)$
- Assuming knowledge of σ^2, and inferred value of δ by observing $\boldsymbol{H} = \text{round}(\boldsymbol{G})$ and $\boldsymbol{y}$
- Used the proposed approximate MLE to estimate $\boldsymbol{x}$

Numerical Experiments

- To validate our method, we solved 10,000 problems. Using the generative model $\boldsymbol{y} = \boldsymbol{G}\boldsymbol{x}_{\text{TRU}} + \boldsymbol{\eta}$ with components drawn as follows:
  - $\boldsymbol{G} \in \mathbb{R}^{m \times n}$, from continuous uniform matrices on $[0, 10]$
  - $\boldsymbol{x}_{\text{TRU}} \in \mathbb{R}^{n}$, taken from the heavy-tailed Cauchy distribution to make the use of prior information on the solution difficult
  - $\boldsymbol{\eta} \in \mathbb{R}^{m}$ has i.i.d. components drawn from a normal distribution, i.e., $\eta_i \sim \mathcal{N}(0, \sigma^2)$
- Assuming knowledge of σ^2, and inferred value of δ by observing $\boldsymbol{H} = \text{round}(\boldsymbol{G})$ and $\boldsymbol{y}$
- Used the proposed approximate MLE to estimate $\boldsymbol{x}$
- Compared to ordinary least squares and total least squares

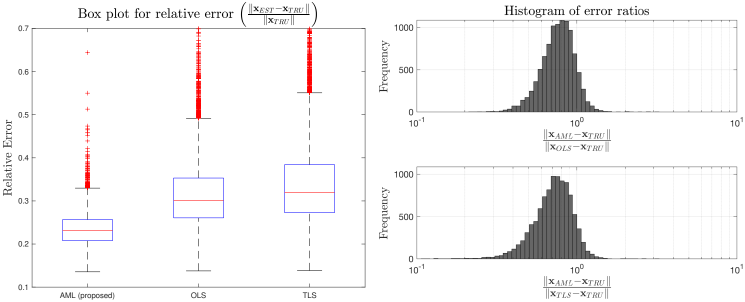

Figure: Error metrics for simulations $\boldsymbol{G} \in \mathbb{R}^{110\times 100}$ and $\sigma = 1$ over $10,000$ simulations. Design matrix rounded to ones spot. Left: box-plot of relative error for different methods. Right: histogram of error ratio $\|\boldsymbol{x}_{\text{AML}} - \boldsymbol{x}_{\text{TRU}}\|/\|\boldsymbol{x}_{\text{OLS}} - \boldsymbol{x}_{\text{TRU}}\|$. Values less than 1 indicate AML outperformed competing method for identical data.

Summary

- Maximum likelihood estimation is useful, but forming likelihood functions for general noise models is difficult

Summary

- Maximum likelihood estimation is useful, but forming likelihood functions for general noise models is difficult
- Presented a method to construct an approximate likelihood function based on MGFs and the saddle point approximation to avoid difficulties

Summary

- Maximum likelihood estimation is useful, but forming likelihood functions for general noise models is difficult
- Presented a method to construct an approximate likelihood function based on MGFs and the saddle point approximation to avoid difficulties
- Found gradient of approximate likelihood using the adjoint state method allowing use of off-the-shelf algorithms

Summary

- Maximum likelihood estimation is useful, but forming likelihood functions for general noise models is difficult
- Presented a method to construct an approximate likelihood function based on MGFs and the saddle point approximation to avoid difficulties
- Found gradient of approximate likelihood using the adjoint state method allowing use of off-the-shelf algorithms
- Showed results of numerical experiments illustrating its effectiveness

References

[1] P. Hall, *The bootstrap and Edgeworth expansion*. Springer Science & Business Media, 2013.

[2] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[3] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *The annals of mathematical statistics*, vol. 27, no. 3, pp. 832–837, 1956.

[4] H. E. Daniels, "Saddlepoint approximations in statistics," *The annals of mathematical statistics*, pp. 631–650, 1954.

[5] R. Lugannani and S. Rice, "Saddle point approximation for the distribution of the sum of independent random variables," *Advances in applied probability*, vol. 12, no. 2, pp. 475–490, 1980.

[6] O. Barndorff-Nielsen and D. R. Cox, "Edgeworth and saddle-point approximations with statistical applications," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 41, no. 3, pp. 279–299, 1979.

[7] G. Strang, *Computational methods for inverse problems*, vol. 23. Wellesley-Cambridge Press, 2007.

[8] J. Nocedal, "Updating quasi-Newton matrices with limited storage," *Math. Comp.*, vol. 25, no. 151, pp. 773–782, 1980.

[9] M. J. Weinstein and A. V. Rao, "Algorithm 984: Adigator, a toolbox for the algorithmic differentiation of mathematical functions in matlab using source transformation via operator overloading," *ACM Transactions on Mathematical Software (TOMS)*, vol. 44, no. 2, p. 21, 2017.

Thank you for your time!

Questions?